HAL
open science

Scheduling coupled-tasks with incompatibility constraint: a bin-packing related problem

Gilles Simonin, Benoit Darties, Rodolphe Giroudeau, Jean-Claude König

To cite this version:

Gilles Simonin, Benoit Darties, Rodolphe Giroudeau, Jean-Claude König. Scheduling coupled-tasks with incompatibility constraint: a bin-packing related problem. BPPC: Bin Packing and Placement Constraints, May 2014, Cork, Ireland. hal-01005429

HAL Id: hal-01005429

https://hal.science/hal-01005429v1

Submitted on 12 Jun 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Scheduling coupled-tasks with incompatibility constraint: a bin-packing related problem

G. Simonin[2], B. Darties[1], R. Giroudeau[3], and J.-C. König[3]

[1] LE2I-CNRS-UMR 6306-8 Rue Alain Savary 21000 Dijon, France
[2] Insight Centre for Data Analytics, University College Cork, Ireland
[3] LIRMM-CNRS-UMR 5506-161, rue Ada 34090 Montpellier, France

Abstract. We tackle the makespan minimization problem of coupled-tasks in presence of compatibility constraint. In particular, we focus on *stretched coupled-tasks*, *i.e.* coupled-tasks having the same sub-tasks execution time and idle time duration. We show the relationship with bin packing problems for some configurations, and study several problems in framework of complexity and approximation for which the topology of the compatibility graph is specific (star, chain, bipartite, ...).

1 Introduction

We consider a non-preemptive coupled-tasks scheduling problem in presence of compatibility constraint on a single processor. In point of view of scheduling theory, the problem is also defined as scheduling problem with exact delays on single machine. In this article, we will show the close relationship between coupled-task in presence of compatibility constraint and the classic bin packing problem in the framework of complexity and approximation.

The coupled-tasks model, was introduced first by Shapiro [8] in order to model some data acquisition processes *i.e.* radar sensors: a sensor emits a radio pulse (first sub-task), and finally listen for an echo reply (second sub-task). Between these two instants (emission and reception), clearly there is idle time due to the propagation, in both sides, of radio pulse. Therefore, a coupled-task is constituted by the triplet: the two sub-tasks and the idle between them. Thus, in order to minimize the makespan (schedule length), it is necessary to execute one or several different sub-tasks during the idle time of a coupled-task. In the basic model, all sub-tasks of coupled-tasks may be executed in each other. Therefore, the aim is to find a best packing of coupled-tasks in which the sum of idle is minimized. Notice that in the basic model, all coupled-tasks may be executed in each other according to processing time of sub-tasks and the duration of the idle time. Hereafter, we consider a relaxation of the previous model in which for a fixed coupled-task $\mathcal{A}$ there are only a subset of coupled-tasks compatible to be processed in the idle time of $\mathcal{A}$. This model is motivated by the problem of data acquisition in presence of compatibility constraint in a submarine torpedo. A collection of sensors acquires data for the torpedo. The compatibility constraint is due to prevent interference issues caused

by tasks using sensors working at the same frequency. So, the constraints are represented by a compatibility graph where vertices are the coupled-tasks and edges represent compatibility between two tasks. In this article the variation of the complexity according to severals structural parameters are considered and some efficient polynomial-time approximation results on $\mathcal{NP}$-hard instances are presented without omitting the relationship with bin packing problems.

Above all, we will show the close relationship between the studied problem and four packing-related problems, for which known approximation will be used as routine for scheduling coupled-tasks problem:

1. The SUBSET SUM (SS) problem: given a set $\mathcal{S}$ of n positive values and $v \in \mathbb{N}$, the aim is to find a subset $\mathcal{S}^* \subseteq \mathcal{S}$ such that $\sum_{i \in \mathcal{S}^*} i = v$. This problem is known to be $\mathcal{NP}$-complete (see [6]). The optimization version problem is sometimes view as a KNAPSACK problem, where each item profits and weights coincide to a value in $\mathcal{S}$, the knapsack capacity is v, and the aim is to find the set of packable items with maximum profit.
2. The MULTIPLE SUBSET SUM (MSS) problem: variant of BIN PACKING in which a number of identical bins is given and one would like to maximize the overall weight of the items packed in the bins in respect with the capacity of each bin. The problem is a special case of the MULTIPLE KNAPSACK problem in which all knapsacks have the same capacity and the item profits and weights coincide. MSS admits a $\mathcal{PTAS}$[2] and a $\frac{3}{4}-$approximation algorithm [3], but does not admit a $\mathcal{FPTAS}$ even for only two knapsacks.
3. MULTIPLE SUBSET SUM WITH DIFFERENT KNAPSACK CAPACITIES (MSSDC) [1] is an extension of MSS considering different bin capacities. MSSDC also admits a $\mathcal{PTAS}$ [1].
4. As a generalization of MSSDC, MULTIPLE KNAPSACK ASSIGNMENT RESTRICTION (MKAR) problem consists in packing weighted items into non-identical capacity-constrained bins, with the additional constraint that each item can be packed into some bins only. Each item as a profit, the objective here is to maximize the sum of profits of packed items. Considering that the profit of each item is equal to its weight, [5] proposed a $\frac{1}{2}$-approximation.

1.1 Coupled-task model in presence of compatibility constraint

We model a task $\mathcal{A}_i$ with a triplet (a_i, L_i, b_i), where a_i (resp. b_i) is the duration of the first (resp. second) sub-task, and L_i the idle time to respect between the execution of sub-tasks. We note $\mathcal{A}$ the set of tasks, and describe the incompatibility constraint between tasks with a graph $G_c = (\mathcal{A}, E)$. There is an edge $(\mathcal{A}_i, \mathcal{A}_j) \in E$ iff a (or both) sub-task from $\mathcal{A}_i$ may be scheduled during the idle time of $\mathcal{A}_j$ or reciprocally. In a valid schedule, we said that $\mathcal{A}_i$ is *packed* into $\mathcal{A}_j$ if the entire task $\mathcal{A}_i$ is scheduled during the idle time of $\mathcal{A}_j$. This is only available when $a_i + L_i + b_i \leq L_j$. We call *stretched coupled-task* a task $\mathcal{A}_i$ such that $a_i = L_i = b_i = \alpha(A_i)$, where $\alpha(A_i)$ is the *stretch factor* of task $\mathcal{A}_i$.

We focus on instances composed with stretched coupled-tasks only, and where two compatible tasks have different stretch factor. In such a configuration, each

edge of G_c can be oriented from the task with the lowest stretch factor to the highest (G_c is a directed acyclic graph). We remark that with stretched coupled-tasks only, when a sub-task of $\mathcal{A}_i$ is executed during the idle time L_j of another one $\mathcal{A}_j$, then the other sub-task of $\mathcal{A}_i$ must also be executed during L_j, and $\mathcal{A}_i$ is packed into $\mathcal{A}_j$, implying also that two compatible tasks $(\mathcal{A}_i, \mathcal{A}_j)$ require $3.\alpha(\mathcal{A}_i) \leq \alpha(\mathcal{A}_j)$.

Using the *three-field* Graham's notation scheme[1] [7], we define the main problem of this study as $1|\alpha(A_i), G_c|C_{max}$. We also introduce $\#(X)$ which counts up the number of different elements in the set X (*i.e.* different values of $\alpha(x)$, for all $x \in X$). In addition, let $d_{G_c}(X)$ be the degree of the task $x \in X$ in G_c, and let $\Delta(G_c)$ be the maximum degree of G_c. Therefore, we present some complexity results according to the values of $\#(X)$ and $\#(Y)$, and propose polynomial-time approximation algorithms results for $\mathcal{NP}$-hard instances.

1.2 From a coupled-task problem to bin packing problem

We focus our study to the instances where G_c describes a bipartite graph $G_c = (X, Y, E)$ with $X \cup Y$ the set of stretched coupled-tasks, and compatibility arcs oriented from X to Y only. On this configuration, the studied scheduling problem is closely related to packing problems:

- Each task $y \in Y$ can be seen as a box with capacity (idle time) of $\alpha(y)$,
- Each task $x \in X$ can be seen as an object, with weight and profit equal to $3.\alpha(x)$. It can fit in certain boxes only, *i.e.* its neighborhood $N(x)$ in G_c.

Our scheduling problem is however different from packing problems in the objective to achieve:

- Packing problems usually consists in maximizing the number (or profit) of packed items;
- On the other way, minimizing the makespan in our scheduling problem corresponds to pack a maximum of tasks from X into tasks from Y, in order to minimize the duration of remaining unpacked tasks, as tasks from Y have to be scheduled sequentially[2]. These remaining tasks are to be scheduled sequentially besides tasks from Y.

There is a relationship between these two objectives, but the well-known results on packing problems would not necessary keep the approximation ratio of algorithms especially designed for packing problems when applied to our scheduling problem. Despite the slightly different objective function, additional conditions on the instance lead to similarities with the following packing problems:

- If $|Y| = 1$ (G_c is a star): a bin-packing version of our problem corresponds to the input of a KNAPSACK problem where each items weight and profit coincide (SS problem), with knapsack of capacity $\alpha(y)$, $y \in Y$, and each items weight coincide to three times the stretch factor of an element in X.

[1] In scheduling theory, a problem type is categorized by its machine environment, job characteristics and objective function.

[2] Y is an independent set for G_c, so tasks from Y must be scheduled sequentially.

- If $|Y| > 1$, $\#(Y) = 1$ and G_c is a complete bipartite graph (contains all arcs from X to Y), the bin-packing version of our problem corresponds to the input of a MULTIPLE KNAPSACK problem in which all knapsacks have the same capacity $\alpha(y)$ such that $y \in Y$, and the item profits and weights coincide to three times the stretch factor of an element in X (also referred to MSS).
- If G_c is a bipartite graph, the bin-packing version corresponds to MKAR with non-identical capacity-constrained bins. Each item x can be packed into bin y iff the arc (x, y) belongs to E.
- If G_c is a bipartite graph and $\#(Y) = 1$, the bin-packing version refers to MKAR with identical capacity-constrained bins.

2 Complexity and approximation results

2.1 Computational complexity results

In this section, we will show the demarcation line between the polynomially solvable and the $\mathcal{NP}$-hardness case according to the several topologies for G_c and different type of stretch length. Due to space limitation, all the proof are omitted here, and are available in the research report dedicated to this work [4].

Theorem 1. *The problem* $1|\alpha(A_i), G_c = chain|C_{max}$ *is polynomial.*

Theorem 2. *The problem* $1|\alpha(A_i), G_c = star|C_{max}$ *is polynomial (resp.* $\mathcal{NP}$*-hard) if the central node admits at least one out-coming arc (resp. admits only incoming arcs).*

Theorem 3. *The problem of deciding whether an instance of* $1|\alpha(A_i)$, $G_c =$*1−stage bipartite*, $\#X \leq n, \#Y \leq n, \Delta_{G_c}(Y) = 2|C_{max}$ *is polynomial.*

Theorem 4. *The problem of deciding whether an instance of* $1|\alpha(A_i), G_c =$ *bipartite*, $\#(X) = 2, \#(Y) = 2, d_{G_c}(x) = 2, d_{G_c}(y) \in \{2, 3\}|C_{max}$ *(resp.* $1|\alpha(A_i)$, $G_c =$ *bipartite*, $\#(X) = 2, \#(Y) = 1, d_{G_c}(x) = \{1, 2\}, d_{G_c}(y) \in \{3, 4\}|C_{max}$ *) is* $\mathcal{NP}$*-hard.*

Theorem 5. *The problem of deciding whether an instance of* $1|\alpha(A_i)$, $G_c =$ *bipartite*, $\#(X) = 1, \#(Y) \leq n|C_{max}$ *is polynomial.*

Corollary 1. *The problem of deciding whether an instance of* $1|\alpha(A_i), \#(X) = 2, \#(Y) = 2, \Delta(G_c) = 2,$ *bipartite*$|C_{max}$ *is polynomial.*

2.2 Polynomial-time approximation algorithms

The challenge for the remaining section, is to propose some efficient polynomial algorithms with a ratio strictly lower than 3/2, which is the ratio returned by a greedy algorithm. We propose a $\mathcal{FPTAS}$ for the star graph whereas some $\mathcal{APX}$-algorithm are developed according to the characteristics of bipartite graph.

Theorem 6. *The problem* $1|\alpha(A_i), G_c = star|C_{max}$ *admits a* $\mathcal{FPTAS}$.

The following Lemma gives approximation preserving reducibility between bin-packing problems and coupled-tasks problems.

Lemma 1. *Let* $\mathcal{P}$ *be a problem with* $\mathcal{P} \in \{\text{MKAR MSSDC}, \text{MSS}\}$ *such that* $\mathcal{P}$ *admits a* ρ*-approximable then* $1|\alpha(A_i), G_c = bipartite|C_{max}$ *(resp.* $1|\alpha(A_i), G_c = complete\ bipartite|C_{max}$ *and* $1|\alpha(A_i), G_c = complete\ bipartite|C_{max}$*, where all the tasks from* Y *have the same stretch factor* $\alpha(y)$ *are approximable to a factor* $1 + \frac{(1-\rho)}{3}$*. The bipartite graph is denoted* (X, Y, E).

Proof. In such instance, any valid schedule consists to find for each task $y \in Y$ a subset of compatible tasks $X_y \subseteq X$ to pack into $y \in Y$, each task of x being packed at most once. Let $X_p = \cup_{y \in Y} X_y$ be the union of tasks of X packed into a task from Y, and $X_{\bar{p}} = X/X_p$. Clearly, we have $seq(X_p) + seq(X_{\bar{p}}) = seq(X)$.

The length of any schedule S is given by the processing time of Y plus the length to schedule sequentially the tasks from $X_{\bar{p}}$. Formally:

$$C_{max}(S) = seq(Y) + seq(X_{\bar{p}}) = seq(Y) + seq(X) - seq(X_p). \tag{1}$$

We use here a reduction to $\mathcal{P}$-problem: ($\forall x \in X$ is an item having a weight $3.\alpha(x)$, and $y \in Y$ is a bin with capacity $\alpha(y)$. Moreover, each item x can be packed on y if and only if the edge $(x, y) \in E$. Using algorithms and results from the literature, one can obtain an assignment of some items into bins, and note X_p the set of packed items. The cost of the solution for the $\mathcal{P}$-problem is $seq(X_p)$. If the $\mathcal{P}$-problem is approximable with a factor ρ, then:

$$seq(X_p) \geq \rho \times seq(X_p^*), C_{max}(S) \leq seq(Y) + seq(X) - \rho \times seq(X_p^*) \tag{2}$$

where X_p^* is the set of packable items with the maximum profit. As X and Y are two fixed sets, a optimal solution S^* with minimal length $C_{max}(S^*)$ is obtained when $seq(X_p)$ is maximum, *i.e.* when $X_p = X_p^*$. The length of any optimal solution is $C_{max}(S^*) = seq(Y) + seq(X) - seq(X_p^*)$ and so

$$\frac{C_{max}(S)}{C_{max}(S^*)} \leq \frac{seq(Y) + seq(X) - \rho \times seq(X_p^*)}{seq(Y) + seq(X) - seq(X_p^*)} \leq 1 + \frac{(1-\rho) \times seq(X_p^*)}{seq(Y) + seq(X) - seq(X_p^*)}$$

By definition, $X_p^* \subseteq X$. Moreover, as the processing time of X_p^* cannot excess the idle time of tasks from Y, we obtain $seq(X_p^*) \leq \frac{1}{3} seq(Y)$. Therefore, we obtain,

$$\rho' = \frac{C_{max}(S)}{C_{max}(S^*)} \leq 1 + \frac{(1-\rho)}{3}. \tag{3}$$

Theorem 7. *Finally, we obtain for coupled-tasks the following results:*

1. $1|\alpha(A_i), G_c = bipartite|C_{max}$ *admits a ratio* $\rho' \leq \frac{7}{6}$ *since* $\rho = \frac{1}{2}$ *for* MKAR*[5].*
2. $1|\alpha(A_i), G_c = complete\ bipartite|C_{max}$ *admits a ratio* $\rho' = 1 + \epsilon$ *since* MSSDC *admits a ratio* $\rho = 1 - \epsilon$ *[1].*
3. $1|\alpha(A_i), G_c = complete\ bipartite|C_{max}$*, where all the tasks from* Y *have the same stretch factor* $\alpha(y), \rho' \leq \frac{13}{12}$ *(resp. admits a* $\rho' = 1+\epsilon$*) since* MSS *admits* $\rho = \frac{3}{4}$ *[3] (resp.* $\rho = 1 - \epsilon$ *[2]).*

3 Conclusion

We investigated a particular coupled-tasks scheduling problem $1|\alpha A_i, G_c|C_{max}$ in presence of a compatibility graph with regard to the complexity and approximation. We established its relationship with packing problem where there is a bipartite compatibility graph, and studied the boundaries between polynomiality and $\mathcal{NP}$-completeness. In such context, we proposed several approximation algorithms. The results announced in this paper are summarized in Table 1.

Topology	**Complexity**	**Approximation**
(G_c)=Chain graph	$O(n^3)$ (Theorem 1)	
(G_c)=Star graph [1]	$\mathcal{NP}-\mathcal{C}$ (Theorem 2)	$\mathcal{FPTAS}$ (Theorem 6)
(G_c)=Star graph [2]	$\mathcal{POLY}$ (Theorem 2)	
G_c= 1-SB, $\Delta(G_c)=2$	$O(n^3)$ (Theorem 3)	
G_c= 1-SC, $\#(X)=2, \#(Y)=2, \Delta_{G_c}=3$	$\mathcal{NP}-\mathcal{C}$ (Theorem 4)	$\frac{7}{6}$-$\mathcal{APX}$(Theorem 7)
G_c= 1-SC,$\#(X)=1, \#(Y)\leq n$	($\mathcal{POLY}$ Theorem 5)	
G_c= 1-SBC	$\mathcal{NP}-\mathcal{C}$ (see [9])	$\mathcal{PTAS}$ (Theorem 7)
G_c= 1-SBC $\#Y=1$	$\mathcal{NP}-\mathcal{C}$ (see [9])	$\mathcal{PTAS}$ (Theorem 7) $\frac{13}{12}$-$\mathcal{APX}$(Theorem 7)

Table 1. Complexity and approximation results. Complete 1-stage bipartite=$1-SBC$

References

1. A. Caprara, H. Kellerer, and U. Pferschy. A PTAS for the Multiple Subset Sum Problem with different knapsack capacities. *Inf. Process. Lett*, 2000.
2. A. Caprara, H. Kellerer, and U. Pferschy. The Multiple Subset Sum Problem. *Siam Journal on Optimization*, 11(2):308–319, 2000.
3. A. Caprara, H. Kellerer, and U. Pferschy. A 3/4-Approximation Algorithm for Multiple Subset Sum. *J. Heuristics*, 9(2):99–111, 2003.
4. A Darties, R Giroudeau, J.-C. König, and G. Simonin. Scheduling stretched coupled-tasks with compatibilities constraints : model, complexity and approximation results for some class of graphs. Technical report, LE2I - University of Burgundy, 2014. http://hal.archives-ouvertes.fr/docs/00/94/75/19/PDF/IR_stretched_tasks.pdf.
5. M. Dawande, J. Kalagnanam, P. Keskinocak, F.S. Salman, and R. Ravi. Approximation Algorithms for the Multiple Knapsack Problem with Assignment Restrictions. *Journal of Combinatorial Optimization*, 4(2):171–186, 2000.
6. M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the theory of NP-completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
7. R L Graham, E L Lawler, J K Lenstra, and A H G Rinnooy Kan. Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. *Annals of Discrete Mathematics*, 5:287–326, 1979.
8. R. D. Shapiro. Scheduling coupled tasks. *Naval Research Logistics Quarterly*, 27:477–481, 1980.
9. G. Simonin, R. Giroudeau, and J.-C. König. Complexity and approximation for scheduling problem for coupled-tasks in presence of compatibility tasks. In *Project Management and Scheduling*. 2010.